Programowanie współbieżne

Laboratorium-7

Fortran 90 - Moduły - Kwaterniony

Moduły

- Kwaterniony. Przeczytaj oryginalny artykuł Hamiltona (**On Quaternions, Sir William Rowan Hamilton**): http://www.maths.tcd.ie/pub/HistMath/People/Hamilton/Quatern2/Quatern2.html
- Utwórz moduł fortranu `Quaternions` pozwalający na dodawanie, mnożenie, itd. kwaternionów. Zastosuj `interface` do definicji operatorów `+, -, *, /` (przeciążanie operatorów). Dodaj funkcję `conjg` (sprzężenie kwaternionu).
- Napisz program korzystający z modułu `Quaternions` i wykonaj proste obliczenia.
- Fragment kodu (następna strona)

Kwaterniony

```
module Quaternions

  public :: quat _print
  private :: quat_add
  intrinsic :: conjg
  public :: conjg

  type, public :: quaternion
     real :: a, b, c, d
  end type quaternion

  interface operator (+)
     module procedure quat_add
  end interface

interface conjg
     module procedure quat_conjg
  end interface

contains

  function quat_add(x,y) result (res)
    type(quaternion), intent(in) :: x, y
    type(quaternion) :: res
    res % a = x % a + y % a
    res % b = x % b + y % b
    res % c = x % c + y % c
    res % d = x % d + y % d
  end function quat_add

  function quat_conjg(x) result (res)
    type(quaternion), intent(in) :: x
    type(quaternion) :: res
    res % a = x % a
    res % b = -(x % b)
    res % c = -(x % c)
    res % d = -(x % d)
  end function quat_conjg

  subroutine quat_print(q)
    type(quaternion), intent(in) :: q
    real :: ap, bp, cp, dp
    ap = q % a
    bp = q % b
    cp = q % c
    dp = q % d
    print "(a1,4f12.6,a1)", "(",ap,bp,cp,dp,")"
  end subroutine quat_print

end module Quaternions
```